EXHIBIT 2

WHITE PAPER

big.LITTLE Technology: The Future of Mobile

Making very high performance available in a mobile envelope without sacrificing energy efficiency

Introduction

With the evolution from the first mobile phones through smartphones to today's superphones and tablets, the demand for compute performance in mobile devices has grown at an incredible rate. Today's devices need to service smarter and more complex interactions, such as voice and gesture control, combined with seamless and reliable content delivery. Gaming and user interfaces have also grown in complexity, with mobile devices now increasingly being used as gaming platforms.

High performance requires fast CPUs which in turn can be difficult to fit in a mobile power or thermal budget. At the same time battery technology has not evolved at the same rate as CPU technology. Therefore today we are in a situation where smartphones require higher performance, but the same power consumption.

The development and design of next generation mobile processors is necessarily guided by the following factors:

1. At the high performance end: high compute capability but within the thermal bounds
2. At the low performance end: very low power consumption

ARM big.LITTLE™ technology has been designed to address these requirements. Big.LITTLE technology is a heterogeneous processing architecture which uses two types of processor. "LITTLE" processors are designed for maximum power efficiency while "big" processors are designed to provide maximum compute performance. Both types of processor are coherent and share the same instruction set architecture (ISA). Using big.LITTLE technology, each task can be dynamically allocated to a big or LITTLE core depending on the instantaneous performance requirement of that task. Through this combination, big.LITTLE technology provides a solution that is capable of delivering the high peak performance demanded by the the latest mobile devices, within the thermal bounds of the system, with maximum energy effiency.

This paper is an overview of the technical aspects of big.LITTLE technology including the hardware components required for a big.LITTLE system, and the software required to manage it.

Same architecture but different micro-architectures

The first big.LITTLE processing pair consists of the ARM Cortex®-A15 and Cortex-A7 processors. Since both processors support the same ARMv7-A ISA, the same instructions or program can be run in a consistent manner on both processors. Differences in the internal microarchitecture of the processors allow them to provide the different power and performance characteristics that are fundamental to the big.LITTLE processing concept. Future designs will also utilise the Cortex-A53 and Cortex-A57 processors in a big.LITTLE implementation.

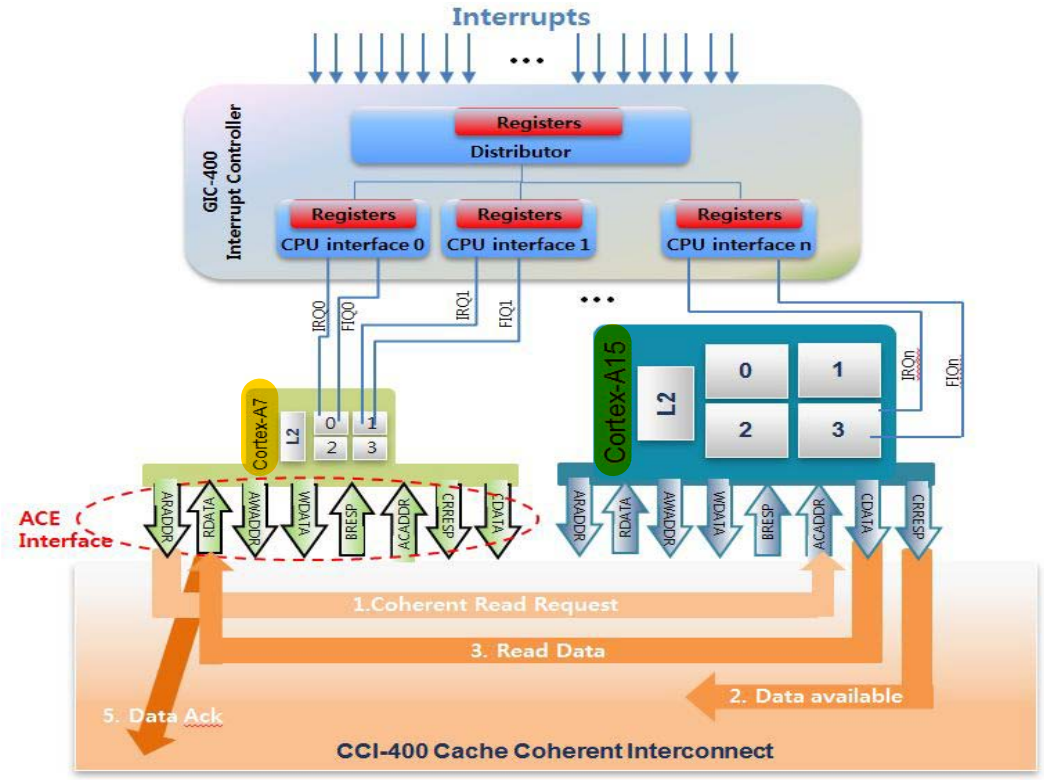

Figure 3: Cache coherency in a big.LITTLE system

As shown in Figure 3, the bus interfaces of Cortex-A15 and Cortex-A7 processors make use of the AMBA® AXI Coherency Extensions (ACE) to the widely-used AMBA AXI protocol. This protocol provides for coherent data transfer at the bus level. In the AMBA ACE protocol, three coherency channels are added in addition to the normal five channels of AMBA AXI. As an example, the lower part of Figure shows the steps in a coherent data read from the Cortex-A7 cluster to the Cortex-A15 cluster. This starts with the Cortex-A7 cluster issuing a Coherent Read Request through the RADDR channel. The CCI-400 hands over the request to the Cortex-A15 processor's ACADDR channel to snoop into Cortex-A15 processor's cache. On receiving the request from CCI-400, the Cortex-A15 processor checks the data availability and reports this information back through the CRRESP channel. If the requested data is in the cache, the Cortex-A15 processor places it on the CDATA channel. Then the CCI-400 moves the data from the Cortex-A15 processor's CDATA channel to the Cortex-A7 processor's RDATA channel, resulting in a cache linefill in the Cortex-A7 processor. The CCI-400 and the ACE protocol enable full coherency between the Cortex-A15 and Cortex-A7 clusters, allowing data sharing to take place without external memory transactions.